DiGA: *Di*stil to *G*eneralize and then *A*dapt for Domain Adaptive Semantic Segmentation

Fengyi Shen[1,2,3], Akhil Gurram[2], Ziyuan Liu[2], He Wang[3,*], Alois C. Knoll[1,*]

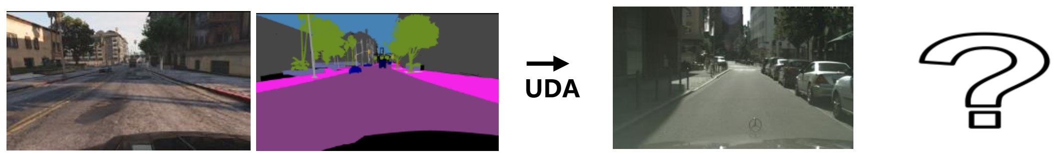
([1]Technische Universität München, [2]Huawei Munich Research Center, [3] EPIC Lab, Peking University)

*corresponding author

CVPR'23 Poster WED-PM-335

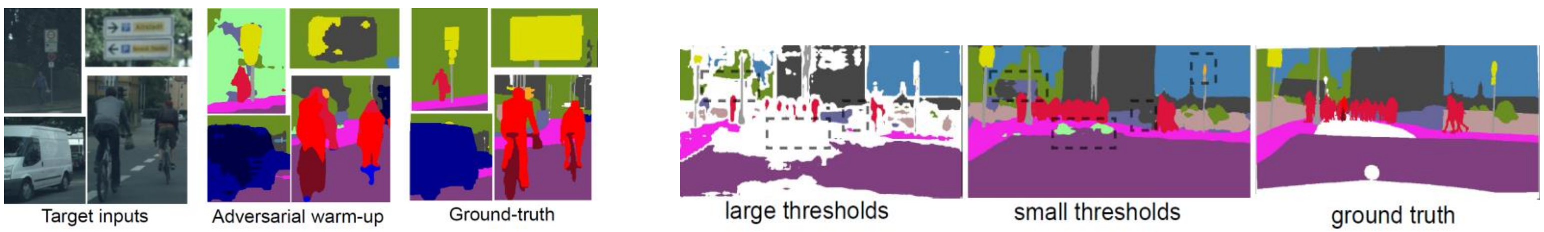
https://github.com/fy-vision/DiGA

Overview

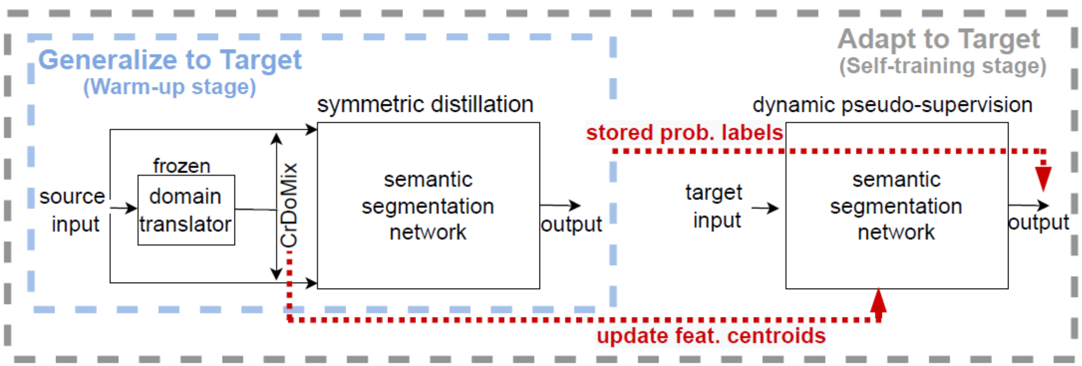


Source domain (labelled)

Target domain (unlabelled)



Stage-wise Training



DiGA Overview

DiGA Demo

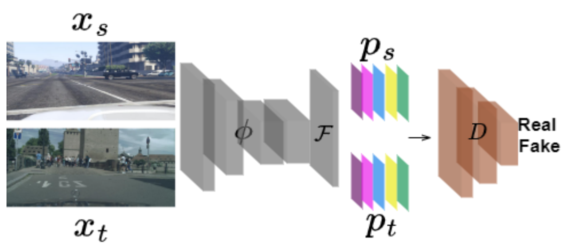
Motivation

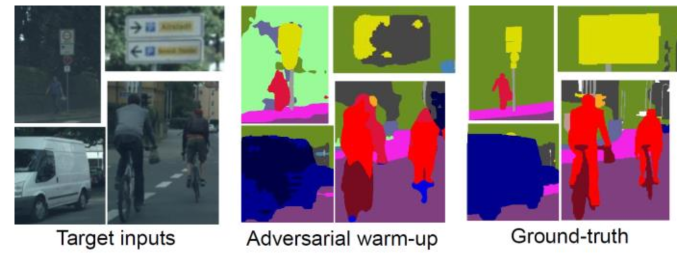
Stage-wise training

Warm-up: Adversarial

AdaptSegNet (CVPR18), BDL (CVPR19)

IAST (ECCV20), ProDA (CVPR21), CPSL (CVPR22) ......





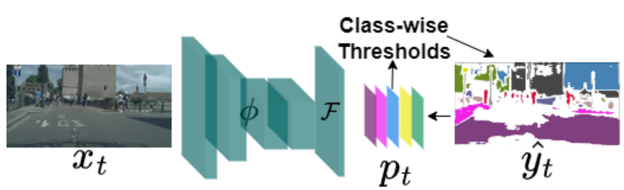
- ***Class-unaware*** X
- ***Blind alignment*** X

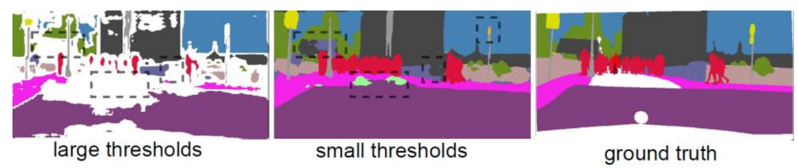
*Question*1: Can the model be trained w/o blindly aligning the target and source features in the warm-up stage?

Self-training (pseudo-label): Thresholds

BDL (CVPR19), CAG (NeurIPS19), IAST (ECCV20), ProDA (CVPR21)

CPSL (CVPR22), BCL (ECCV22), ProCA (ECCV22) ......





- ***Large: insufficient learning*** X
- ***Small: introduce noise*** X

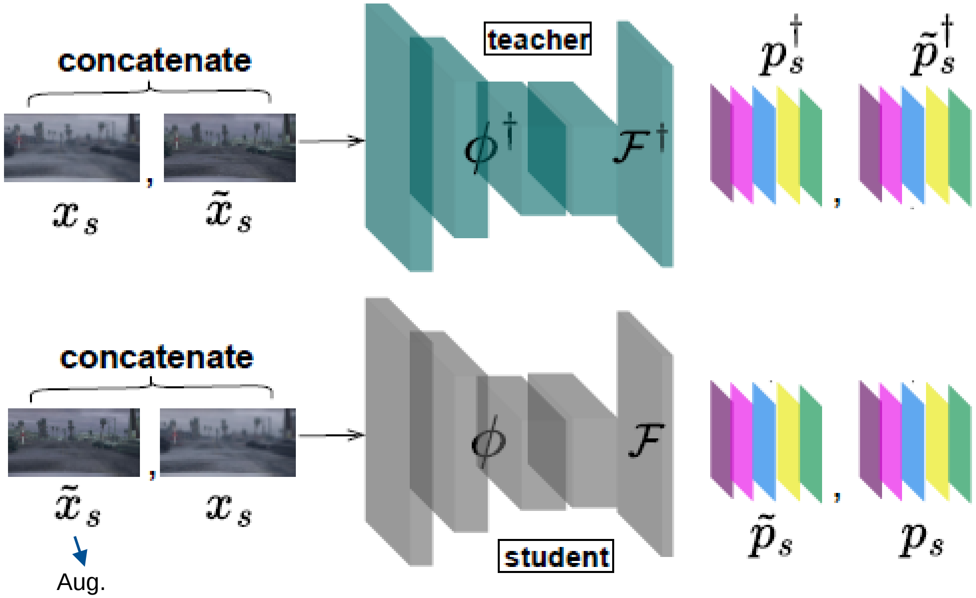
*Question*2: Is it possible to avoid looking for thresholds during the self-training stage?

Method (Warm-up Stage)

Pixel-wise symmetric knowledge distillation:

$$\{p_s^\dagger,\ \tilde{p}_s^\dagger\} = \sigma(\mathcal{F}^\dagger \circ \phi^\dagger(\{x_s,\ \tilde{x}_s\})) \quad (2)$$

$$\{\tilde{p}_s,\ p_s\} = \sigma(\mathcal{F} \circ \phi(\{\tilde{x}_s,\ x_s\})) \quad (3)$$

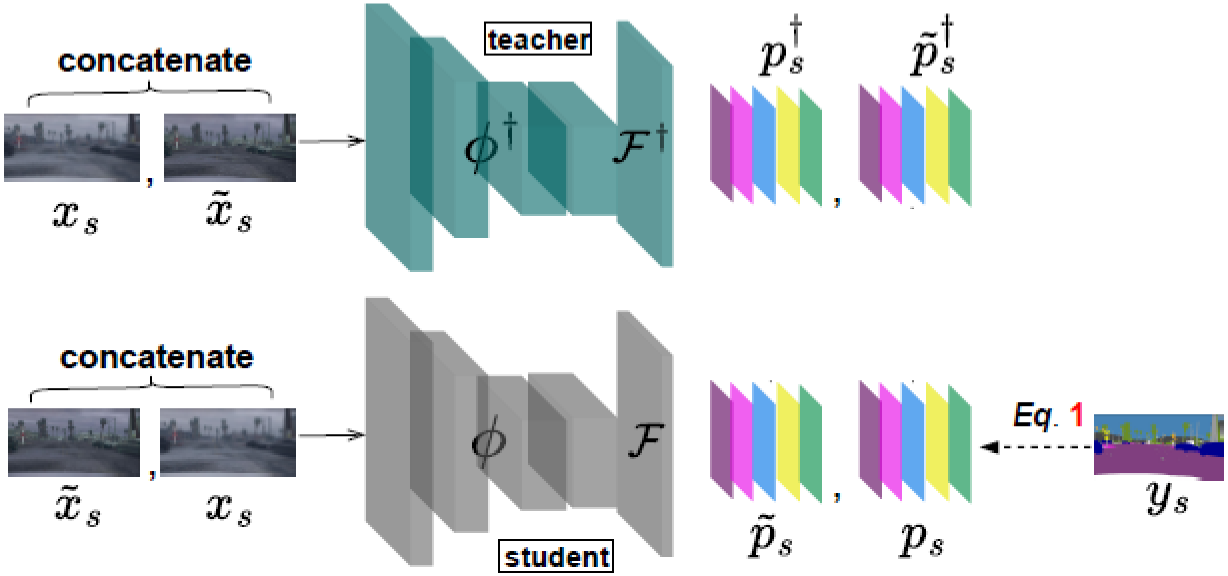


Pixel-wise symmetric knowledge distillation:

$$\mathcal{L}_s^{seg} = \sum_{h,w} \sum_{c} -y_s^{(c,h,w)} \log(p_s)^{(c,h,w)} \quad (1)$$

$$\{p_s^{\dagger},\ \tilde{p}_s^{\dagger}\} = \sigma(\mathcal{F}^{\dagger} \circ \phi^{\dagger}(\{x_s,\ \tilde{x}_s\})) \quad (2)$$

$$\{\tilde{p}_s,\ p_s\} = \sigma(\mathcal{F} \circ \phi(\{\tilde{x}_s,\ x_s\})) \quad (3)$$



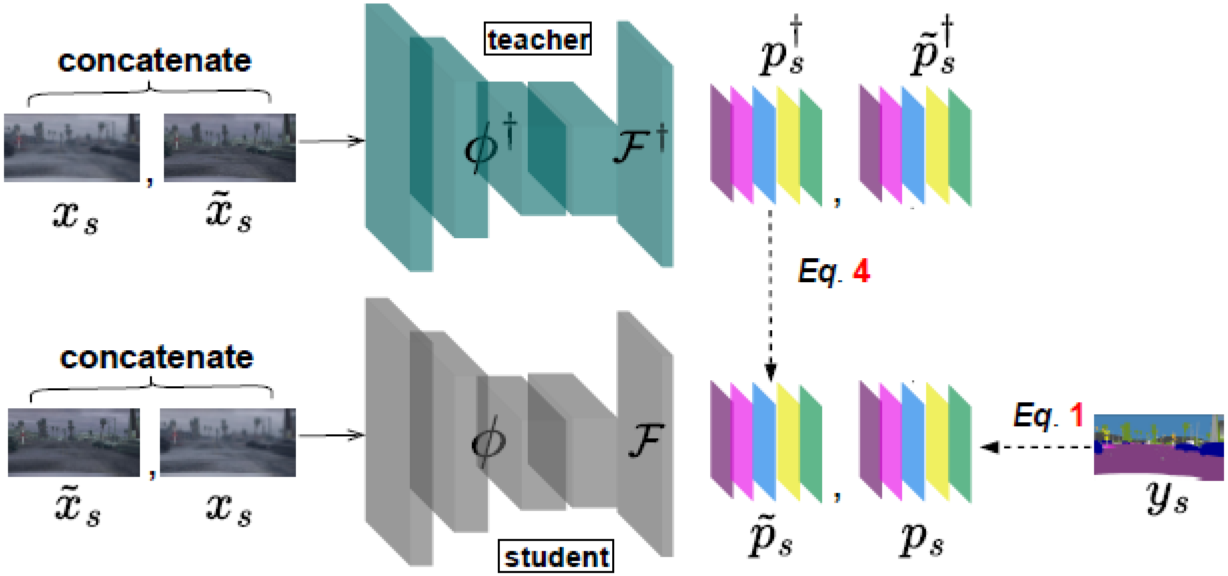
Method (Warm-up Stage)

Pixel-wise symmetric knowledge distillation:

$$\mathcal{L}_s^{seg} = \sum_{h,w} \sum_{c} -y_s^{(c,h,w)} \log(p_s)^{(c,h,w)} \quad (1)$$

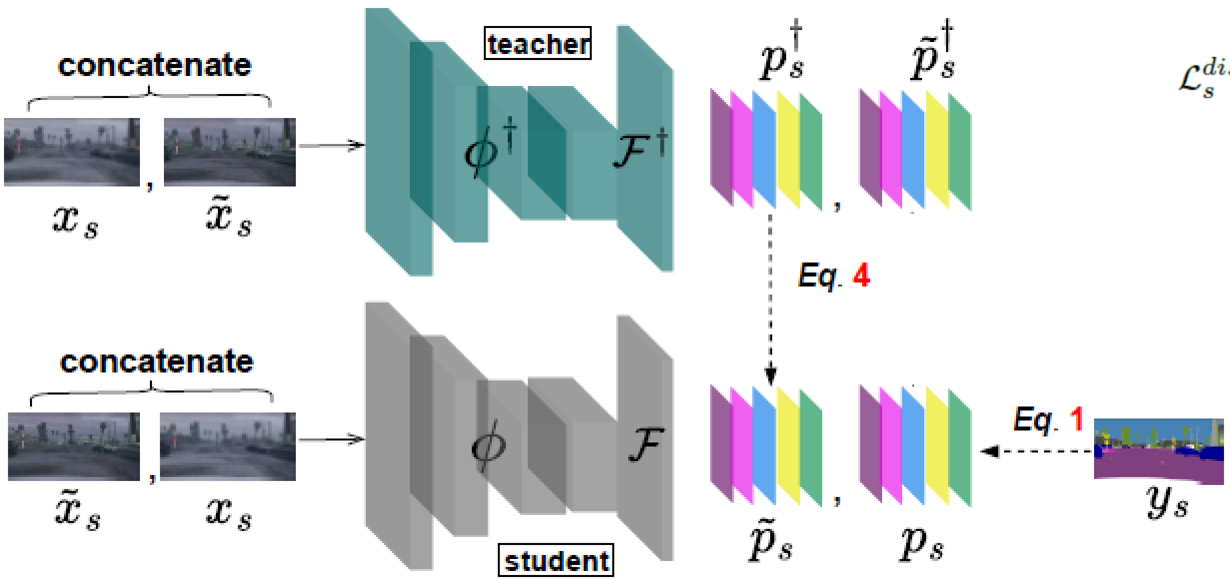
$$\{p_s^\dagger,\ \tilde{p}_s^\dagger\} = \sigma(\mathcal{F}^\dagger \circ \phi^\dagger(\{x_s,\ \tilde{x}_s\})) \quad (2)$$

$$\{\tilde{p}_s,\ p_s\} = \sigma(\mathcal{F} \circ \phi(\{\tilde{x}_s,\ x_s\})) \quad (3)$$



Method (Warm-up Stage)

Pixel-wise symmetric knowledge distillation:



$$\mathcal{L}_s^{seg} = \sum_{h,w}\sum_{c} -y_s^{(c,h,w)} \log(p_s)^{(c,h,w)} \tag{1}$$

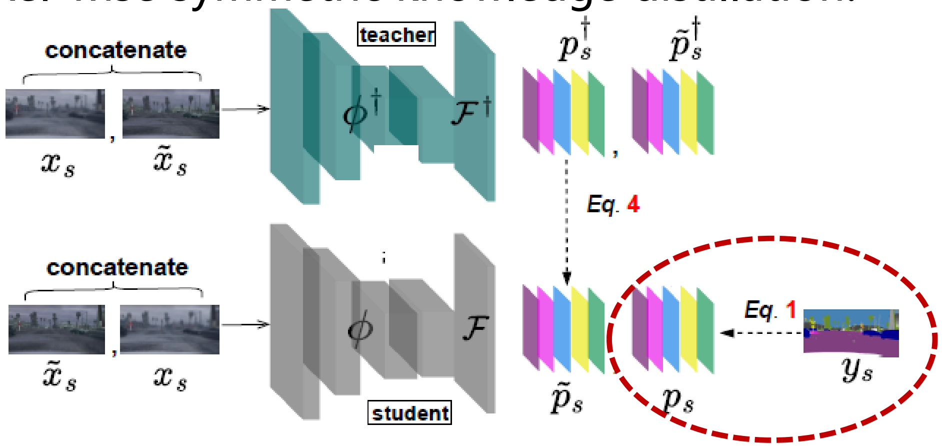
$$\{p_s^\dagger,\ \tilde{p}_s^\dagger\} = \sigma(\mathcal{F}^\dagger \circ \phi^\dagger(\{x_s,\ \tilde{x}_s\})) \tag{2}$$

$$\{\tilde{p}_s,\ p_s\} = \sigma(\mathcal{F} \circ \phi(\{\tilde{x}_s,\ x_s\})) \tag{3}$$

$$\mathcal{L}_s^{distil} = \overline{\mathcal{H}(p_s^\dagger, \tilde{p}_s)} \tag{4}$$

where $\mathcal{H}(a,b) = -a\log(b)$

Pixel-wise symmetric knowledge distillation:



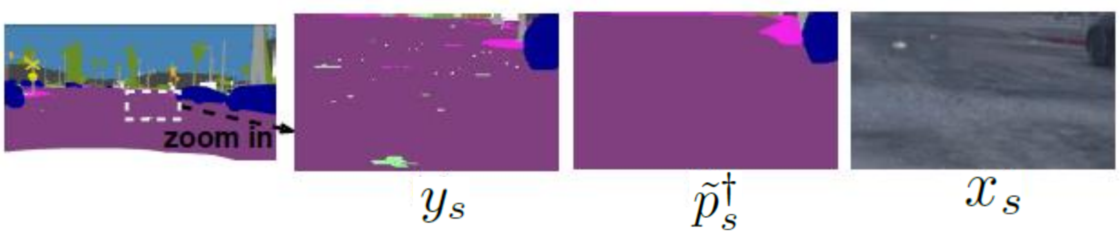
$$\mathcal{L}_s^{seg} = \sum_{h,w} \sum_{c} -y_s^{(c,h,w)} \log(p_s)^{(c,h,w)} \quad (1)$$

$$\{p_s^\dagger,\ \tilde{p}_s^\dagger\} = \sigma(\mathcal{F}^\dagger \circ \phi^\dagger(\{x_s,\ \tilde{x}_s\})) \quad (2)$$

$$\{\tilde{p}_s,\ p_s\} = \sigma(\mathcal{F} \circ \phi(\{\tilde{x}_s,\ x_s\})) \quad (3)$$

$$\mathcal{L}_s^{distil} = \overline{\mathcal{H}(p_s^\dagger, \tilde{p}_s)} \quad (4)$$

where $\mathcal{H}(a,b) = -a \log(b)$



Not all source labels are useful for adaptation!

Method (Warm-up Stage)

Pixel-wise symmetric knowledge distillation:

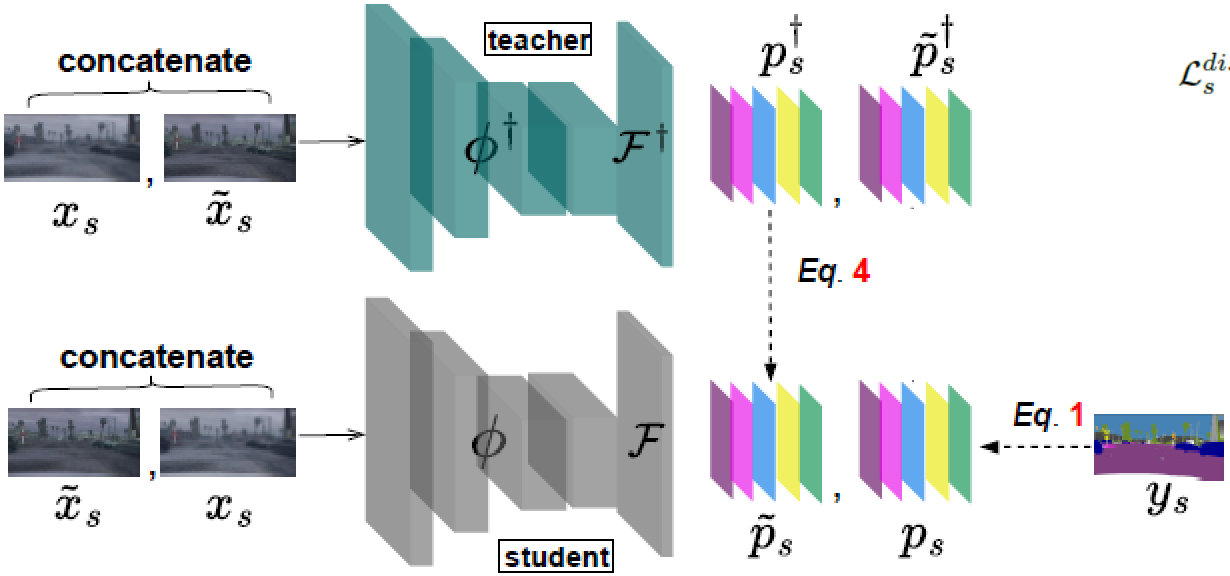
$$\mathcal{L}_s^{seg} = \sum_{h,w} \sum_{c} -y_s^{(c,h,w)} \log(p_s)^{(c,h,w)} \tag{1}$$

$$\{p_s^{\dagger},\ \tilde{p}_s^{\dagger}\} = \sigma(\mathcal{F}^{\dagger} \circ \phi^{\dagger}(\{x_s,\ \tilde{x}_s\})) \tag{2}$$

$$\{\tilde{p}_s,\ p_s\} = \sigma(\mathcal{F} \circ \phi(\{\tilde{x}_s,\ x_s\})) \tag{3}$$

$$\mathcal{L}_s^{distil} = \overline{\mathcal{H}(p_s^{\dagger}, \tilde{p}_s)} \tag{4}$$

where $\mathcal{H}(a,b) = -a\log(b)$



Method (Warm-up Stage)

Pixel-wise symmetric knowledge distillation:

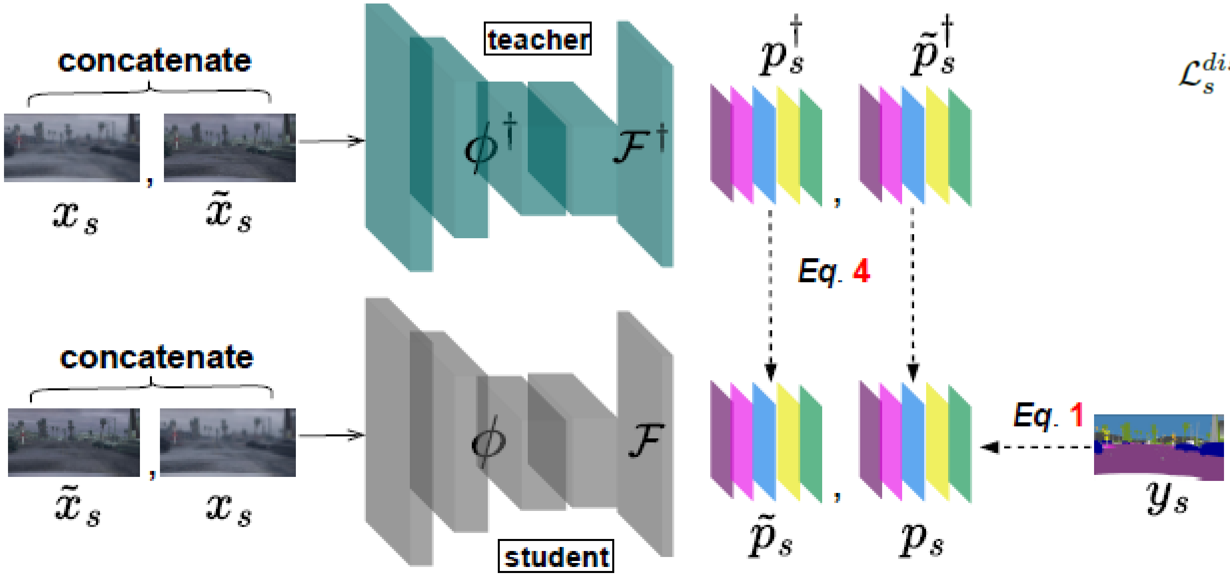
$$\mathcal{L}_s^{seg} = \sum_{h,w} \sum_{c} -y_s^{(c,h,w)} \log(p_s)^{(c,h,w)} \tag{1}$$

$$\{p_s^\dagger,\ \tilde{p}_s^\dagger\} = \sigma(\mathcal{F}^\dagger \circ \phi^\dagger(\{x_s,\ \tilde{x}_s\})) \tag{2}$$

$$\{\tilde{p}_s,\ p_s\} = \sigma(\mathcal{F} \circ \phi(\{\tilde{x}_s,\ x_s\})) \tag{3}$$

$$\mathcal{L}_s^{distil} = \overline{\mathcal{H}(p_s^\dagger, \tilde{p}_s)} + \alpha \overline{\mathcal{H}(\tilde{p}_s^\dagger, p_s)} \tag{4}$$

where $\mathcal{H}(a,b) = -a \log(b)$



Method (Warm-up Stage)

Pixel-wise symmetric knowledge distillation:

$$\mathcal{L}_s^{seg} = \sum_{h,w}\sum_{c} -y_s^{(c,h,w)} \log(p_s)^{(c,h,w)} \tag{1}$$

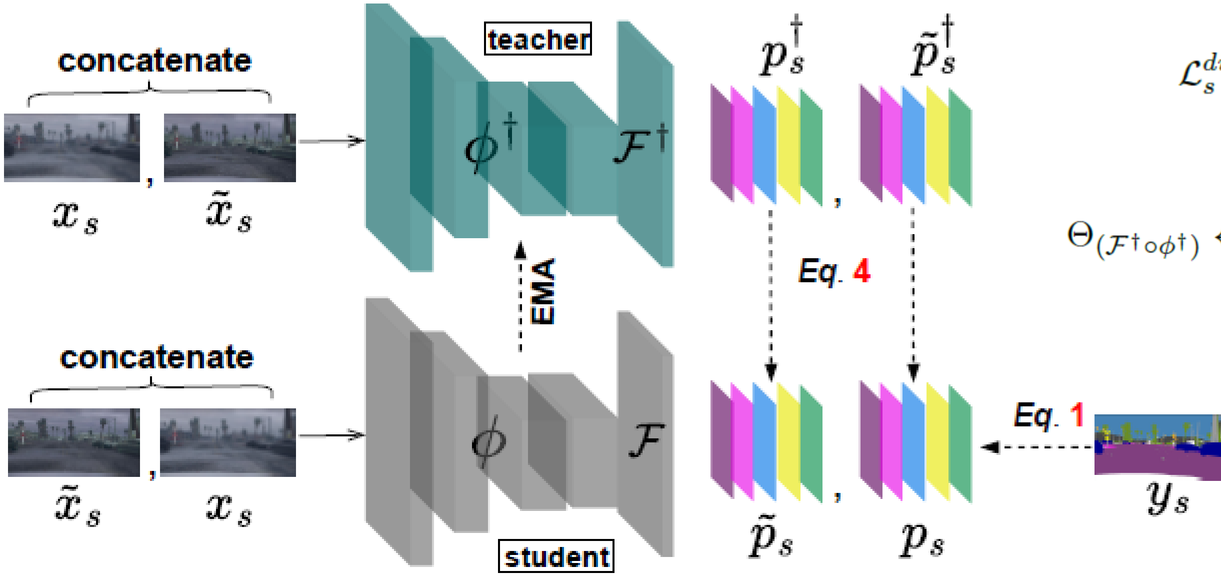
$$\{p_s^\dagger,\ \tilde{p}_s^\dagger\} = \sigma(\mathcal{F}^\dagger \circ \phi^\dagger(\{x_s,\ \tilde{x}_s\})) \tag{2}$$

$$\{\tilde{p}_s,\ p_s\} = \sigma(\mathcal{F} \circ \phi(\{\tilde{x}_s,\ x_s\})) \tag{3}$$

$$\mathcal{L}_s^{distil} = \overline{\mathcal{H}(p_s^\dagger, \tilde{p}_s)} + \alpha\overline{\mathcal{H}(\tilde{p}_s^\dagger, p_s)} \tag{4}$$

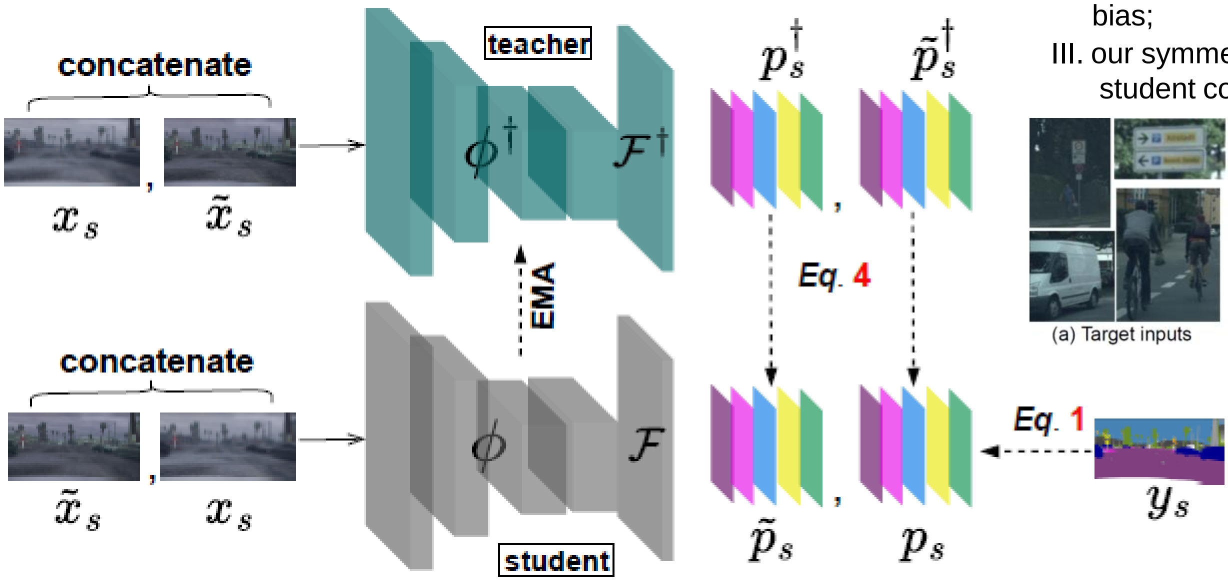
where $\mathcal{H}(a,b) = -a\log(b)$

$$\Theta_{(\mathcal{F}^\dagger\circ\phi^\dagger)} \leftarrow \xi * \Theta_{(\mathcal{F}^\dagger\circ\phi^\dagger)} + (1-\xi) * \Theta_{(\mathcal{F}\circ\phi)} \tag{5}$$



Method (Warm-up Stage)

Pixel-wise symmetric knowledge distillation:



(+) Benefits are threefold:

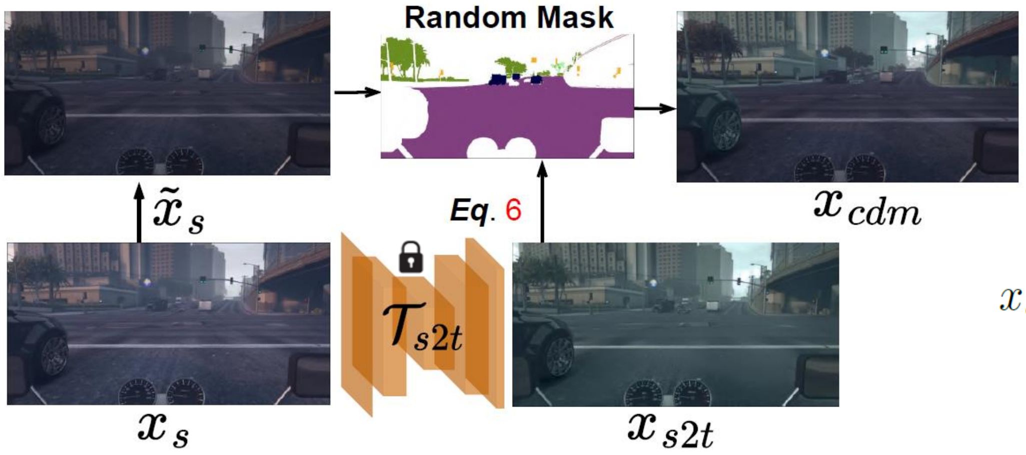
I. knowledge distillation only on source domain, the learning becomes class-aware;

II. soft labels avoid the model overfitting to domain-specific bias;

III. our symmetric proposal ensures bidirectional teacher-student consistency between different inputs.

(a) Target inputs (b) Adversarial warm-up (c) Our warm-up (d) Ground-truth

Cross-Domain Mixture Data Augmentation:



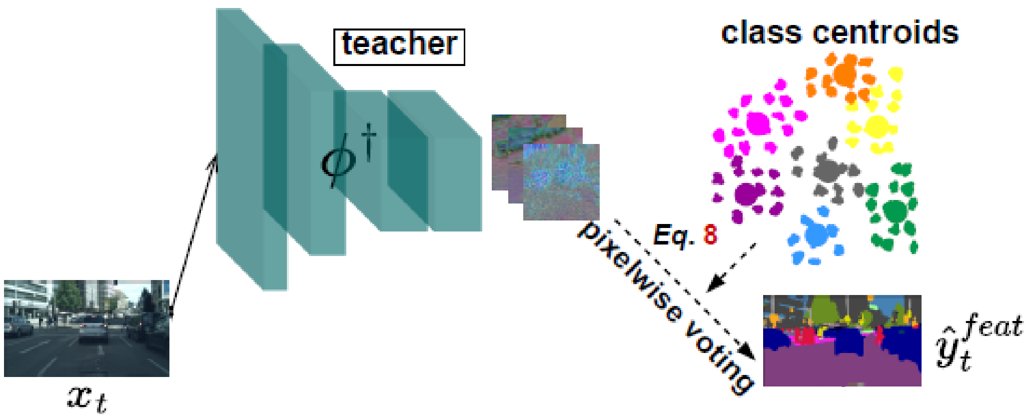
$$x_{cdm} = \tilde{x}_s \odot \mathcal{M} + x_{s2t} \odot (1 - \mathcal{M}) \tag{6}$$

(+) Benefits :

I. adding both OOD and target-aware information into data augmentation, distillation more meaningful;

II. carrying multiple effects on a single view w/o increasing batch size;

Bilateral-consensus Pseudo-supervision:

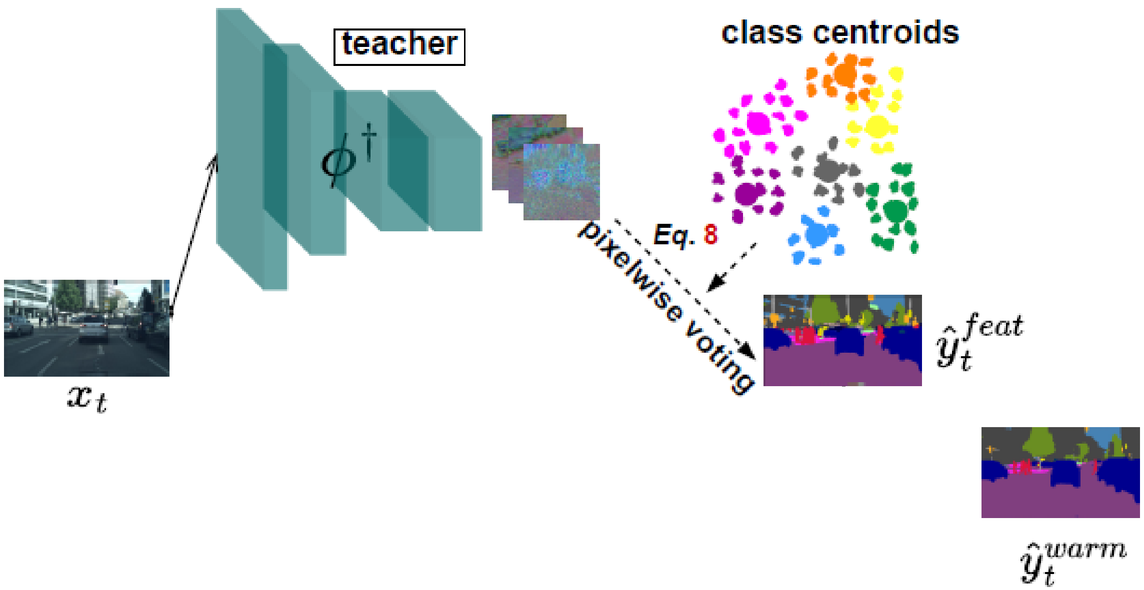


$$\Lambda_s = \{\rho^{(k)}, k = 1, 2..., c\}$$

$$\rho^{(k)} = \frac{\sum_{N_s} GAP(\phi(x_{cdm})^{(k)} \odot (y_s^{(k)} = 1))}{\sum_{N_s} \mathbb{1} \odot (y_s^{(k)} = 1)} \quad (7)$$

$$\hat{y}_t^{feat(jk)} = \mathcal{O}(\arg\min ||\phi^{\dagger}(x_t)^{(jk)} - \Lambda||_2) \quad (8)$$

Bilateral-consensus Pseudo-supervision:



$$\Lambda_s = \{\rho^{(k)}, k = 1, 2..., c\}$$

$$\rho^{(k)} = \frac{\sum_{N_s} GAP(\phi(x_{cdm})^{(k)} \odot (y_s^{(k)} = 1))}{\sum_{N_s} \mathbb{1} \odot (y_s^{(k)} = 1)} \quad (7)$$

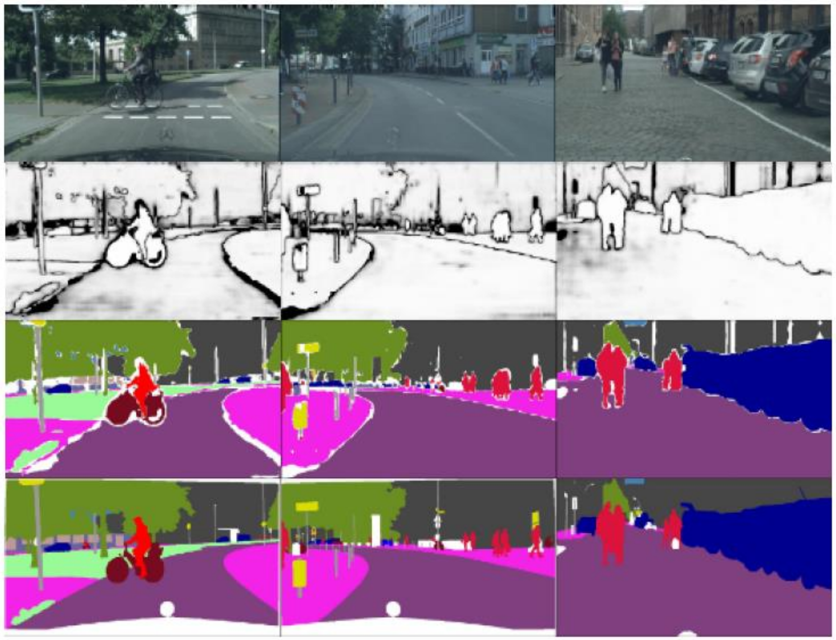
$$\hat{y}_t^{feat(jk)} = \mathcal{O}(\arg\min ||\phi^{\dagger}(x_t)^{(jk)} - \Lambda||_2) \quad (8)$$

Method (Self-training Stage)

$\hat{y}_t^{feat}$

$\hat{y}_t^{warm}$

$$\hat{y}_t = \hat{y}_t^{feat} \cap \hat{y}_t^{warm}$$



prediction uncertainty
(darker means lower confidence)

pseudo-label

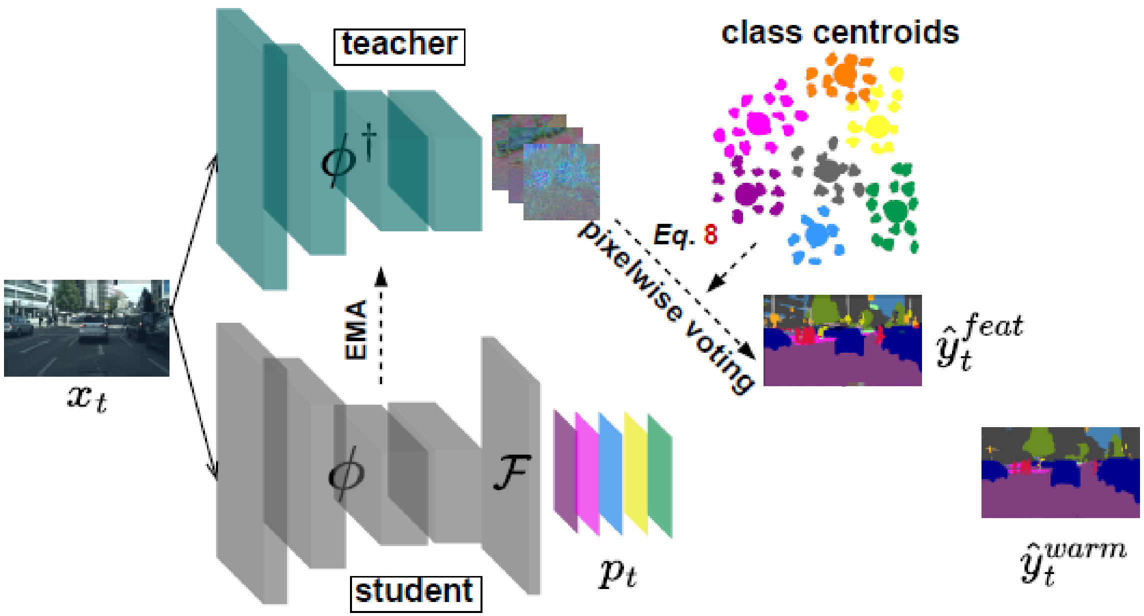
ground-truth
(unused in training)

Bilateral-consensus Pseudo-supervision:

$$\Lambda_s = \{\rho^{(k)}, k = 1, 2..., c\}$$

$$\rho^{(k)} = \frac{\sum_{N_s} GAP(\phi(x_{cdm})^{(k)} \odot (y_s^{(k)} = 1))}{\sum_{N_s} \mathbb{1} \odot (y_s^{(k)} = 1)} \quad (7)$$

$$\hat{y}_t^{feat(jk)} = \mathcal{O}(\arg\min ||\phi^{\dagger}(x_t)^{(jk)} - \Lambda||_2) \quad (8)$$



Bilateral-consensus Pseudo-supervision:

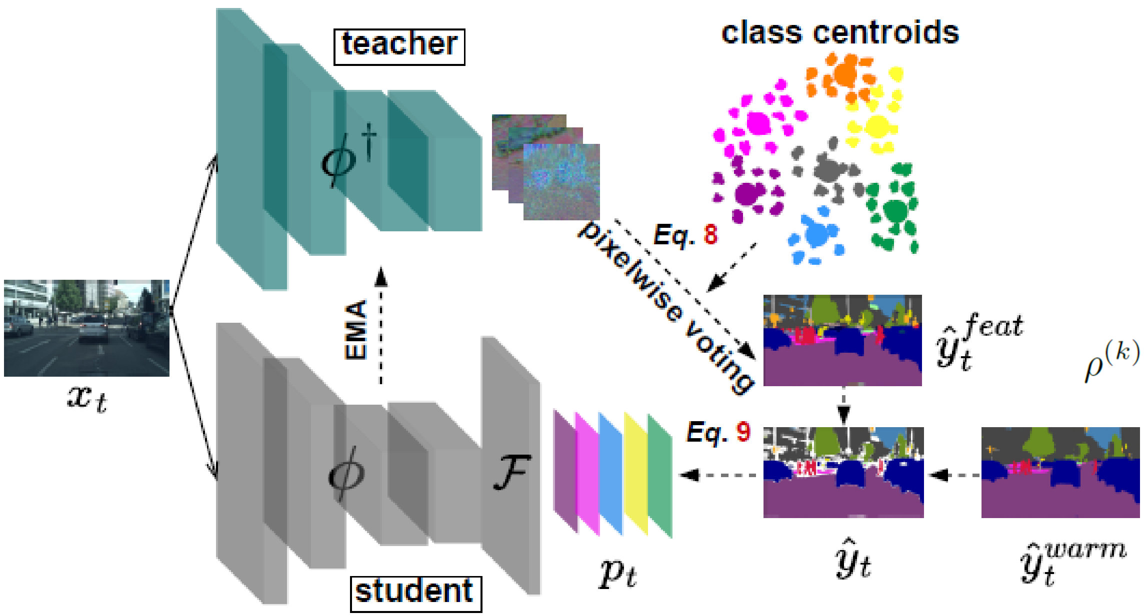
$$\Lambda_s = \{\rho^{(k)}, k = 1, 2..., c\}$$

$$\rho^{(k)} = \frac{\sum_{N_s} GAP(\phi(x_{cdm})^{(k)} \odot (y_s^{(k)} = 1))}{\sum_{N_s} \mathbb{1} \odot (y_s^{(k)} = 1)} \quad (7)$$

$$\hat{y}_t^{feat(jk)} = \mathcal{O}(\arg\min ||\phi^{\dagger}(x_t)^{(jk)} - \Lambda||_2) \quad (8)$$

$$\hat{\mathcal{L}}_t^{seg} = \sum_{h,w}\sum_{c} -\hat{y}_t^{(c,h,w)} \log(p_t)^{(c,h,w)} \quad (9)$$

$$\rho^{(k)} \leftarrow \delta(\delta\rho^{(k)} + (1-\delta)\rho_s'^{(k)}) + (1-\delta)\rho_t'^{(k)} \quad (10)$$



(+) Benefits :

I. threshold-free self-training;
II. dynamic pseudo-label selection;

$$\mathcal{L}_{DiGA} = \lambda_s^{distil}\mathcal{L}_s^{distil} + \lambda^{seg}(\mathcal{L}_s^{seg} + \hat{\mathcal{L}}_t^{seg}) \quad (11)$$

Experiments (Quantitative Evaluation)

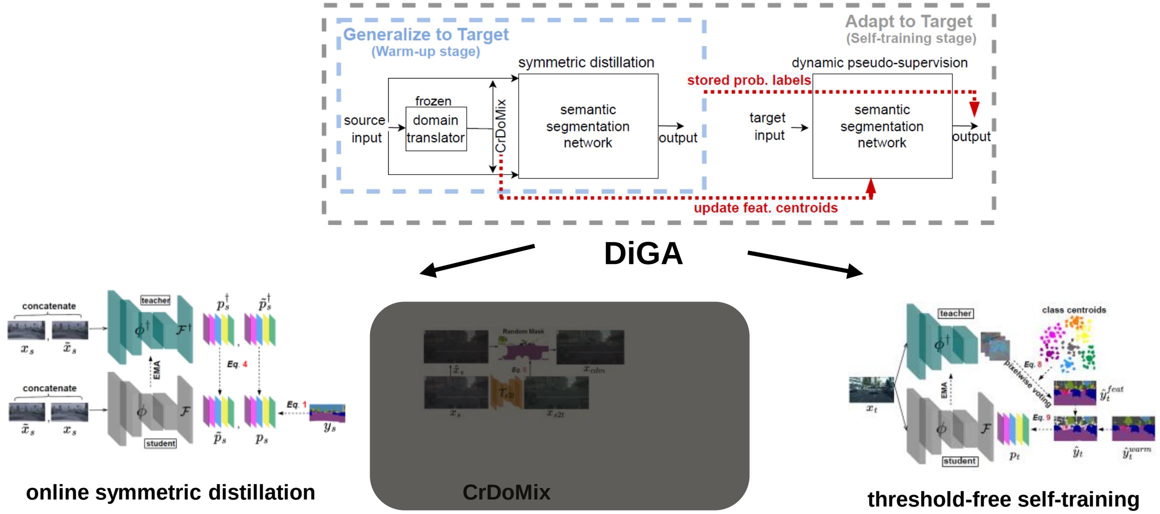
Method	road	sdwk	bldng	wall	fence	pole	light	sign	veg	trrn	sky	psn	rider	car	truck	bus	train	moto	bike	mIoU
BDL [41]	91.0	44.7	84.2	34.6	27.6	30.2	36.0	36.0	85.0	43.6	83.0	58.6	31.6	83.3	35.3	49.7	3.3	28.8	35.6	48.5
ProDA‡ [80]	91.5	52.4	82.9	42.0	35.7	40.0	44.4	43.3	**87.0**	**43.8**	79.5	66.5	31.4	86.7	41.1	52.5	0.0	45.4	53.8	53.7
CPSL‡ [40]	91.7	52.9	83.6	43.0	32.3	43.7	51.3	42.8	85.4	37.6	81.1	69.5	30.0	88.1	44.1	59.9	24.9	47.2	48.4	55.7
ProCA [31]	91.9	48.4	87.3	41.5	31.8	41.9	47.9	36.7	86.5	42.3	84.7	68.4	43.1	88.1	39.6	48.8	**40.6**	43.6	56.9	56.3
DiGA (Ours, ResNet)	**95.6**	**67.4**	**89.8**	**51.6**	**38.1**	**52.0**	**59.0**	**51.5**	86.4	34.5	**87.7**	**75.6**	**48.8**	**92.5**	**66.5**	**63.8**	19.7	**49.6**	**61.6**	**62.7**
DiGA (Ours, HRNet)	95.2	65.2	90.7	59.0	57.1	57.8	63.3	54.8	90.0	42.4	89.0	76.8	49.6	91.6	66.8	69.8	59.7	24.0	51.9	66.1
DAFormer [27]	95.7	70.2	89.4	53.5	**48.1**	49.6	55.8	59.4	89.9	47.9	**92.5**	72.2	44.7	92.3	74.5	78.2	65.1	55.9	61.8	68.3
DiGA (Ours + DAFormer)	95.7	**70.4**	**89.8**	**54.8**	47.8	**51.3**	**57.8**	**63.9**	**90.3**	**48.8**	91.8	**73.1**	**46.6**	**92.6**	**78.5**	**81.3**	**74.8**	**57.3**	**63.2**	**70.0**
HRDA [28]	96.4	74.4	91.0	**61.6**	51.5	**57.1**	63.9	69.3	91.3	48.4	**94.2**	79.0	52.9	**93.9**	**84.1**	85.7	75.9	**63.9**	**67.5**	73.8
DiGA (Ours + HRDA)	**97.0**	**78.6**	**91.3**	60.8	**56.7**	56.5	**64.4**	**69.9**	**91.5**	**50.8**	93.7	**79.2**	**55.2**	93.7	78.3	**86.9**	**77.8**	63.7	65.8	**74.3**

Table 1. **GTA5-to-Cityscapes adaptation results**. We compare our model performance with state-of-the-art methods. In all tables of Sec. 4.2, bold stands for **best** and underline for second-best. ‡ for fair comparison, we use their reported results after ST stage.

Method	road	sdwk	bldng	wall*	fence*	pole*	light	sign	veg	sky	psn	rider	car	bus	mcycl	bcycl	mIoU	mIoU*
BDL [41]	86.0	46.7	80.3	-	-	-	14.1	11.6	79.2	81.3	54.1	27.9	73.7	42.2	25.7	45.3	-	51.4
ProDA‡ [80]	87.1	44.0	83.2	26.9	0.0	42.0	45.8	34.2	86.7	81.3	68.4	22.1	87.7	50.0	31.4	38.6	51.9	58.5
CPSL‡ [40]	87.3	44.4	83.8	25.0	0.4	42.9	47.5	32.4	86.5	83.3	69.6	29.1	89.4	52.1	42.6	54.1	54.4	61.7
ProCA [31]	90.5	52.1	84.6	**29.2**	**3.3**	40.3	37.4	27.3	86.4	**85.9**	69.8	28.7	88.7	53.7	14.8	54.8	53.0	59.6
DiGA (Ours, ResNet)	89.1	**53.4**	**86.1**	28.7	3.0	**49.6**	**50.6**	**34.9**	**88.2**	84.9	**71.3**	**40.9**	**91.6**	**75.1**	**50.3**	**65.8**	**60.2**	**67.9**
DiGA (Ours, HRNet)	90.6	56.3	87.4	38.8	6.4	57.7	59.3	50.4	87.9	86.4	76.1	47.9	89.0	54.2	47.2	69.1	62.8	69.4
DAFormer [27]	84.5	40.7	**88.4**	41.5	6.5	50.0	55.0	**54.6**	86.0	89.8	73.2	48.2	87.2	53.2	53.9	61.7	60.9	67.4
DiGA (Ours + DAFormer)	**85.2**	**41.4**	88.2	**42.6**	**7.5**	**52.1**	**57.5**	47.7	**87.8**	**90.8**	**75.0**	**50.8**	**87.8**	**58.0**	**58.5**	**63.0**	**62.1**	**68.6**
HRDA [28]	85.2	47.7	88.8	49.5	4.8	**57.2**	**65.7**	60.9	85.3	92.9	**79.4**	**52.8**	89.0	**64.7**	**63.9**	64.9	65.8	72.4
DiGA (Ours + HRDA)	**88.5**	**49.9**	**90.1**	**51.4**	**6.6**	55.3	64.8	**62.7**	**88.2**	**93.5**	78.6	51.8	**89.5**	62.2	61.0	**65.8**	**66.2**	**72.8**

Table 2. **Synthia-to-Cityscapes adaptation results**. mIoU, mIoU* refer to 16-class and 13-class experimental settings, respectively. ‡ for fair comparison, we use their reported results after ST stage following [31].

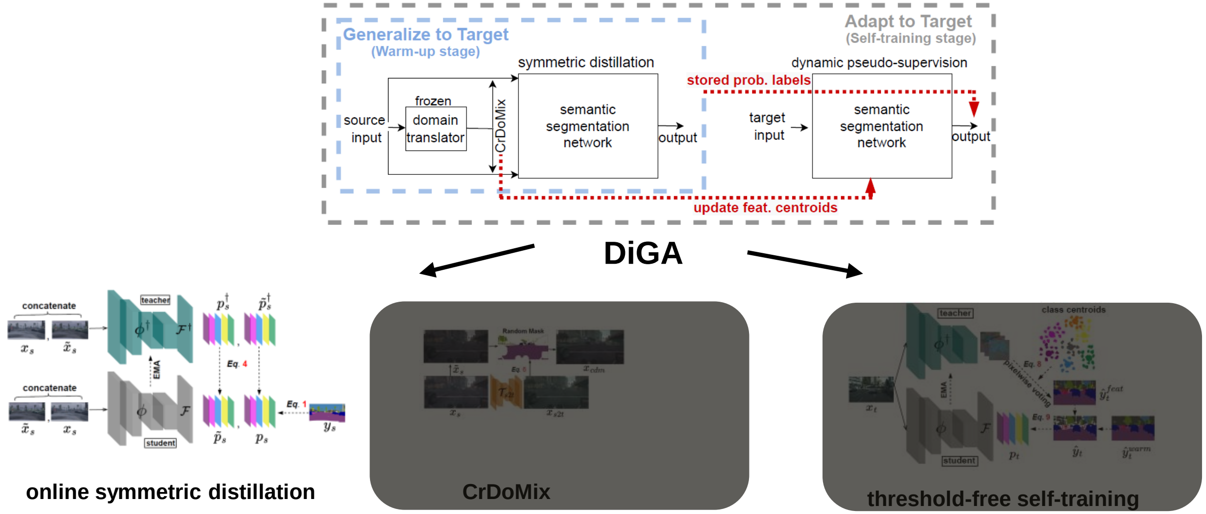
Experiments (Semi-supervised Semantic Segmentation)



Method	Cityscapes			
	1/16 (186)	1/8 (372)	1/4 (744)	1/2 (1488)
CPS [9]	75.09	77.92	79.24	80.67
Ours	**76.86**	**78.51**	**80.01**	**80.93**

Table 7. **mIoU comparison of semi-supervised semantic segmentation** using HRNet backbone, based on which SOTA performance of CPS [9] is reported. Evaluation performed on Cityscapes validation set under different partition protocols.

Experiments (Domain Generalization)



Method	Train on GTA5 (G)			
	→C	→B	→M	→S
ISW [12]	42.87	38.53	39.05	29.58
SFDA [71]	43.50	-	-	-
SAN-SAW [56]	45.33	41.18	40.77	31.84
SHADE [83]	46.66	43.66	45.50	-
Our Distillation	**48.87**	**44.42**	**51.78**	**37.17**

Table 6. **mIoU comparison with SOTA methods for domain generalization.** G, C, B, M and S denote GTA5, Cityscapes, BDD100k, Mapillary and Synthia, respectively. For fair comparison, all the listed methods

Stage-wise:

Method	a	b	c	d	e	mIoU	Δ
Source-only						38.3	+0.0
Source-only	✓					38.9	+0.6
(i)	✓	✓				46.7	+8.4
(ii)	✓	✓	✓			48.9	+10.6
(iii)	✓	✓	✓	✓		51.1	+12.8
(iv)	✓	✓	✓	✓	✓	62.7	+24.4

Table 3. **DiGA components**: **a** → MST, **b** → $\overline{\mathcal{H}(p_s^\dagger, \tilde{p}_s)}$, **c** → $\overline{\mathcal{H}(\tilde{p}_s^\dagger, p_s)}$, **d** → CrDoMix, and **e** → $\hat{\mathcal{L}}_t^{seg}$.

Warm-up stage:

Strategy	Adv. [67]	Distil.	Adv. [67]+CrDoMix	Distil.+CrDoMix
mIoU	45.2	48.9	47.3	**51.1**

Table 4. **Warm-up model comparison** between adversarial training and our knowledge distillation w/ and w/o CrDoMix.

Self-training stage:

Strategy	(1)$\hat{y}_t^{feat}$	(2)$\hat{y}_t^{warm}$	(3)BDL	(4)ProDA	(5)$\hat{y}_t(ours)$
mIoU	52.1	53.8	56.2	59.5	**62.7**

Table 5. mIoU comparison of applying different pseudo-labelling techniques to train ST stage **based on our warm-up model**.

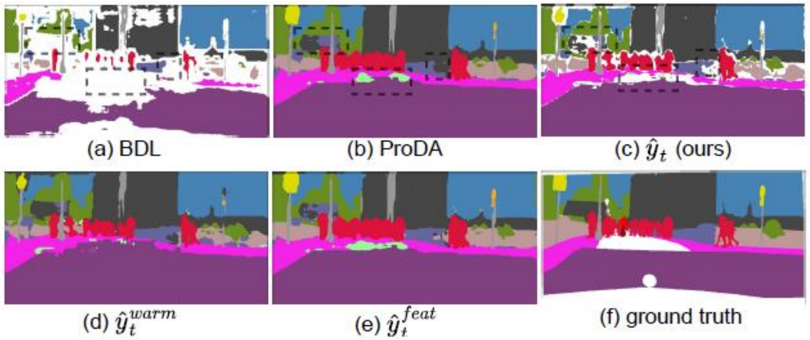
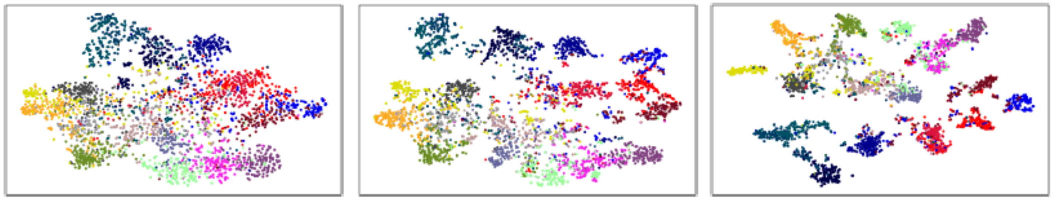


Figure 7. **Comparison of different pseudo-labelling techniques** given the same input image, and ground truth (f) is only adopted for comparison. Dashed black boxes reveal the major differences.



(a) source-only (b) warm-up stage (c) ST stage

Figure 6. **Visualization of feature distribution** on Cityscapes validation set for each stage based on t-SNE [18] map.

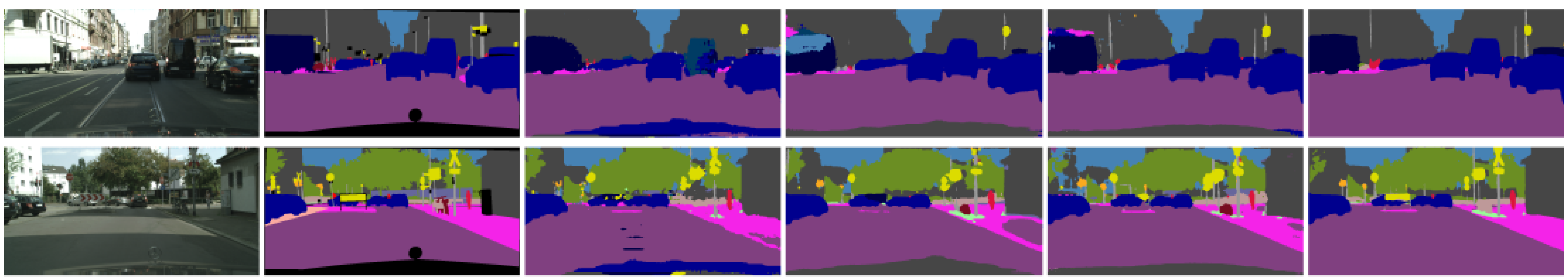


Figure 6. **Qualitative results of GTA5-to-Cityscapes adaptation on Cityscapes validation set**. Columns from left to right are: target domain inputs; ground-truth labels; segmentation predictions of BDL [41], ProDA [80], CPSL [40] and DiGA (`ResNet`).